Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 1 (2012) 2417–2423
Procedia Computer Science
www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2010

L_p-norm proximal support vector machine and its applications

Wen Jing Chen[1], Ying Jie Tian*

Research Center on Fictitious Economy and Data Science, CAS, Beijing, China

Abstract

In this paper, we propose an efficient l_p-norm $(0 < p < 1)$Proximal Support Vector Machine by combining proximal support vector machine (PSVM) and feature selection strategy together. Following two lower bounds for the absolute value of nonzero entries in every local optimal solution of the model, we investigate the relationship between sparsity of the solution and the choice of the regularization parameter and p-norm. After smoothing the problem in l_p-norm PSVM, we solved it by smoothing conjugate gradient method (SCG) method, and preliminary numerical experiments show the effectiveness of our model for identifying nonzero entries in numerical solutions and feature selection, and we also apply it to a real-life credit dataset to prove its efficiency.

Keywords: proximal support vector machine, feature selection, smoothing conjugate gradient, l_p norm

1. Introduction

Support vector machine (SVM) [1, 2] is a promising tool in machine learning, but it is unable to get the importance feature. To identifying a subset of features which contribute most to classification is also an important task in classification. The benefit of feature selection is twofold. It leads to parsimonious models that are often preferred in many scientific problems, and it is also crucial for achieving good classification accuracy in the presence of redundant features[3, 4, 16, 18]. We can combine SVM with various feature selection strategies, Some of them are "filters": general feature selection methods independent of SVM. That is, these methods select important features first and then SVM is applied for classification. On the other hand, some are wrapper-type methods: modifications of SVM which choose important features as well as conduct training/testing. In the machine learning literature, there are several proposals for feature selection to accomplish the goal of automatic feature selection in the SVM[4]-[9], in some of which they applied the l_0-norm, l_1-norm or l_∞-norm SVM and got competitive performance.

Naturally, we expect that using the l_p-norm $(0 < p < 1)$ in SVM can find more sparse solution than using l_1-norm. [10] considered a minimization model where the objective function is the sum of a data fitting term in l_2 norm and a regularization term in l_p norm $(0 < p < 1)$, and gave out several interesting theoretical results. Because the problem they solved is formulated as

$$\min_{x \in R^n} \quad \|Ax - b\|_2^2 + \lambda \|x\|_p^p, \tag{1}$$

* *Email address:* `tianyingjie1213@163.com,Corresponding author` (Ying Jie Tian)
[1]This work has been partially supported by grants from National Natural Science Foundation of China(NO.10601064, NO.70621001)

doi:10.1016/j.procs.2010.04.272

, where $A \in R^{m\times n}, b \in R^m, \lambda \geq 0, 0 < p < 1$, therefore in this paper we naturally want to combine proximal support vector machine (PSVM) [12] and feature selection strategy by introducing the l_p-norm $(0 < p < 1)$. Because the primal problem in PSVM can be directly transformed to formulation (1), so that we can apply the theoretical results in [10] to PSVM and study its corresponding properties. This paper is organized as follows. In Section 2 we propose the l_p-norm proximal support vector machine for classification, and based on two lower bounds in [10], we present the corresponding lower bounds for the absolute value of nonzero entries in every local optimal solution of our new model. In Section 3, we smooth the problem in l_p-norm PSVM and introducing the conjugate gradient method (SCG) to solve it. Section 4 investigates the performance of l_p-norm PSVM, and compare it with other feature selection methods by numerical experimental on UCI datasets, and also apply it to a real-life credit dataset to prove its efficiency. Finally, we have discussion and conclusions in Section 4.

2. Lower bounds for nonzero entries in solutions of l_p-norm Proximal SVM

For a classification problem, the training set is given by

$$T = \{(x_1, y_1), ..., (x_l, y_l)\} \in (R^n \times \{-1, 1\})^l, \tag{2}$$

where $x_i = ([x_i]_1, \cdots, [x_i]_n)^{\mathrm{T}} \in R^n$ and $y_i \in \{-1, 1\}, i = 1, \cdots, l$. Proximal support vector machine aims to build a decision function by solving the following primal problem:

$$\min_{w,b,\eta} \quad \frac{1}{2}(\|w\|^2 + b^2) + \frac{C}{2}\sum_{i=1}^{l}\eta_i^2, \tag{3}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) = 1 - \eta_i, i = 1, \cdots, l, \tag{4}$$

in which $\|w\|^2$ is the l_2-norm of w. Figure 1 describe its geometric explanation in R^2, the planes $(w \cdot x) + b = \pm 1$ around which points of the points "∘" and points "+" cluster and which are pushed apart by the optimization problem (3)~(4). PSVM leads to an extremely fast and simple algorithm for generating a linear or nonlinear classifier that merely requires the solution of a single system of linear equations, and has been efficiently applied to many fields.

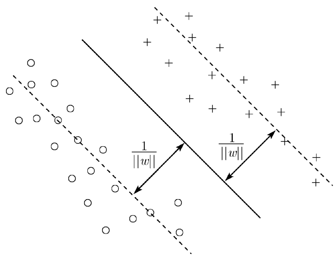

figure1

In order to get more sparse solutions, we substitute the l_2-norm by l_p-norm$(0 < 1 < p)$ in problem (3)~(4) and it turns to be

$$\min_{w,b,\eta} \quad \lambda(\|w\|_p^p + |b|^p) + \sum_{i=1}^{l}\eta_i^2, \tag{5}$$

$$\text{s.t.} \quad y_i((w \cdot x_i) + b) = 1 - \eta_i, i = 1, \cdots, l, \tag{6}$$

in which

$$\|w\|_p^p = \sum_{i=1}^{n}|w_i|^p. \tag{7}$$

Obviously, problem (5)~(6) is equivalent to the following unconstrained minimization problem

$$\min_{z\in R^{n+1}} \|Az-e\|_2^2+\lambda\|z\|_p^p, \tag{8}$$

where

$$z = (w^{\mathrm{T}}, b)^{\mathrm{T}} \in R^{n+1}, \tag{9}$$

$$A = \begin{pmatrix} y_1x_1^{\mathrm{T}}, y_1 \\ \ddots \\ y_lx_l^{\mathrm{T}}, y_l \end{pmatrix} \in R^{l\times(n+1)}, \tag{10}$$

$$e = (1,\cdots,1)^{\mathrm{T}} \in R^l. \tag{11}$$

we call problem (8) l_p-norm PSVM problem. Paper [10] established two lower bounds for the absolute value of nonzero entries in every local optimal solution of the general model (8), which can be used to eliminate zero entries precisely in any numerical solution. Therefor, we apply them to describe the performance for feature selection of our l_p-norm PSVM for classification.

Let $\mathcal{Z}_p^*$ denote the set of local solutions of problem , then for any $z^*\in\mathcal{Z}_p^*$, we have the corresponding versions of theorem 2.1 and theorem 2.2 in [10] for model (8):

Theorem 2.1(first bound) Let $L=(\frac{\lambda p}{2\beta})^{\frac{1}{1-p}}$, where $\beta=\|A\|\|e\|$, then for any $z^*\in\mathcal{Z}_p^*$, we have

$$z_i^*\in(-L,L)\Rightarrow z_i^*=0,\ i=1,\cdots,n+1. \tag{12}$$

Theorem 2.2(second bound) Let $L_i=(\frac{\lambda p(1-p)}{2\|a_i\|^2})^{\frac{1}{2-p}}, i=1,\cdots,n+1$, where a_i is the ith column of the matrix A (10), then for any $z^*\in\mathcal{Z}_p^*$, we have

$$z_i^*\in(-L_i,L_i)\Rightarrow z_i^*=0,\ i=1,\cdots,n+1. \tag{13}$$

Just as pointed out by [10], the above two theorems clearly shows the relationship between the sparsity of the solution and the choice of the regularization parameter λ and norm $\|\cdot\|_p$. The lower bounds is not only useful for identification of zero entries in all local optimal solutions from approximation ones, but also for selection of the regularization parameter λ and norm p.

3. Smoothing l_p-norm PSVM problem

However, solving the nonconvex, non-Lipschitz continuous minimization problem (8) is very dicult. Most optimization algorithms are only efficient for smooth and convex problems, and they can only find local optimal solutions. In [10], they first smooth problem (8) by choosing appropriate smoothing function and then apply the smoothing conjugate gradient method (SCG)[11] to solve it, which guarantees that any accumulation point of a sequence generated by this method is a Clarke stationary point of the nonsmooth and nonconvex optimization problem (8). Here we introduce the smoothing function and then smooth problem (8). Let $s_\mu(\cdot)$ be a smoothing function of $|t|$, which takes the formulations as

$$s_\mu(t)=\begin{cases} |t|, & \text{if } |t|>\frac{\mu}{2}, \\ \frac{t^2}{\mu}+\frac{\mu}{4}, & \text{if } |t|\le\frac{\mu}{2}, \end{cases} \tag{14}$$

so the smoothed problem (8) is

$$\min_{z\in R^{n+1}} \|Az-e\|_2^2+\lambda\sum_{i=1}^{n+1}(s_\mu(z_i))^p, \tag{15}$$

Therefore, based on solving (15), and determining the nonzero elements of solutions by two bounds, we can establish the l_p-norm PSVM algorithm for both feature selection and classification problem:

Algorithm 1. (l_p-PSVM)

(1) Given a training set $T = \{(x_1, y_1), ..., (x_l, y_l)\} \in (R^n \times \{-1, 1\})^l$;
(2) Select appropriate parameters λ and p;
(3) Solve problem (15) using SCG method and get the solution $z^* = (w^{*\mathrm{T}}, b^*)^{\mathrm{T}}$;
(4) Set the variables w_i^* to zero if it satisfies either of the two bounds, get the sparse solution w^*;
(5) Select the features corresponding to nonzero elements of w^*;
(6) Construct the decision function as

$$f(x) = \mathrm{sgn}((w^* \cdot x) + b^*). \tag{16}$$

4. Numerical experiments and application

In this section, based on several UCI datasets, we first apply model (8) to investigate the performance of feature selection by the choice of λ and norm $\|\cdot\|_p$. And because [3][9] has compared the l_1-norm SVM, l_2-norm SVM and l_∞-norm, we then only conduct the issue of numerically comparing svm with l_1-norm with our l_p-method. Before doing experiments, data sets are scaled with each feature to [0, 1]. At last we apply algorithm 1 to a real-life credit dataset. The computational results are conducted on a Dell PC (1.80 GHz, 1.80 GHz, 512MB of RAM) with using Matlab 7.4.

4.1. Relationship between the sparsity of solutions with λ and p

For every dataset, we choose parameter $\lambda \in \{2^0, 2^1, \cdots, 2^7\}$ and $p \in [0.1, 0.9]$ with step 0.1. The following tables describe the sparsity of solutions of problem (8) under corresponding (λ, p), and $(\sharp_1, \sharp_2)$ in each table means the number of zero variables in solutions w^* determined by bound (12) and bound (13) separately. Each Bold number denotes the maximum number for a given λ and varying p.

Table 1: Experiments On Heart Disease Dataset (l=270, n=13).

$\lambda \setminus p$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
2^0	(0, 0)	(0, 0)	(0, **1**)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
2^1	(0, 0)	(0, **1**)	(0, 0)	(0, 0)	(0, **1**)	(0, **1**)	(0, 0)	(0, 0)	(0, 0)
2^2	(0, 0)	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, 0)	(0, 0)
2^3	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)
2^4	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)
2^5	(0, 2)	(0, **3**)	(0, **3**)	(0, **3**)	(0, **3**)	(0, **3**)	(0, **3**)	(0, 2)	(0, 1)
2^6	(0, 7)	(0, 7)	(0, **8**)	(0, **8**)	(0, **8**)	(0, 6)	(0, 4)	(0, 4)	(0, 3)
2^7	(0, 9)	(0, 10)	(0, **11**)	(0, **11**)	(0, **11**)	(0, **11**)	(0, 8)	(0, 7)	(0, 5)

From all the four tables we can see that: bound (13) gives out more sparsity than bound (12); for bound (13), the sparsity value takes its maximum mainly at $p \in (0.2, 0.6)$ for any given λ, which also can be roughly estimated by

$$p^*(\lambda) = arg \max_{0<p<1} (\lambda p(1-p))^{1/(2-p)} \tag{17}$$

for $\lambda \in (0, 2^7)$ if we scale a_i such that $\|a_i\| = 1$; the number of nonzero entries in any local minimizer of (8) reduces when λ becomes larger.

Table 2: Experiments On German Credit Dataset (l=1000, n=24)

$\lambda \setminus p$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
2^0	(0, 1)	(0, **2**)	(0, 1)	(0, 1)	(0, **2**)	(0, 1)	(0, 1)	(0, 1)	(0, 1)
2^1	(0, **2**)	(0, **2**)	(0, **2**)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 0)	(0, 0)
2^2	(0, 2)	(0, **3**)	(0, 2)	(0, 2)	(0, 1)	(0, 2)	(0, 1)	(0, 1)	(0, 1)
2^3	(0, 3)	(0, **5**)	(0, 4)	(0, 2)	(0, 2)	(0, 2)	(0, 1)	(0, 1)	(0, 0)
2^4	(0, **5**)	(0, **5**)	(0, **5**)	(0, **5**)	(0, 4)	(0, 3)	(0, 2)	(0, 1)	(0, 1)
2^5	(0, **7**)	(0, **7**)	(0, 6)	(0, **7**)	(0, **7**)	(0, 6)	(0, 4)	(0, 1)	(0, 1)
2^6	(1, 7)	(0, **10**)	(0, **10**)	(0, 9)	(0, 9)	(0, 7)	(0, 7)	(0, 3)	(0, 1)
2^7	(1, 12)	(1, **13**)	(0, **13**)	(0, **13**)	(0, 12)	(0, 10)	(0, 9)	(0, 6)	(0, 2)

Table 3: Experiments On Australian credit Dataset (l=690, n=14)

$\lambda \setminus p$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
2^0	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, 0)	(0, **1**)	(0, 0)
2^1	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, **1**)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
2^2	(0, 1)	(0, 1)	(0, **2**)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 1)	(0, 0)
2^3	(0, 1)	(0, 1)	(0, **2**)	(0, 1)	(0, **2**)	(0, 1)	(0, 1)	(0, 1)	(0, 0)
2^4	(0, 3)	(0, 4)	(0, **5**))	(0, 4)	(0, 2)	(0, 1)	(0, 1)	(0, 1)	(0, 1)
2^5	(1, 5)	(0, **6**)	(1, **6**)	(0, **6**)	(0, **6**)	(0, 5)	(0, 2)	(0, 2)	(0, 1)
2^6	(0, **6**)	(0, **6**)	(0, **6**)	(0, **6**)	(0, **6**)	(0, **6**)	(0, **6**)	(0, 3)	(0, 2)
2^7	(0, 7)	(1, 9)	(1, 9)	(0, **10**)	(0, **10**)	(0, 8)	(0, 7)	(0, 6)	(0, 3)

Table 4: Experiments On Sonar Dataset (l=208, n=60)

$\lambda \setminus p$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
2^0	(0, 1)	(0, **5**)	(0, 2)	(0, 1)	(0, 3)	(0, 1)	(0, 0)	(0, 0)	(0, 0)
2^1	(0, 2)	(0, **4**)	(0, 3)	(0, 2)	(0, 2)	(0, 2)	(0, 0)	(0, 4)	(0, 0)
2^2	(0, 11)	(0, **9**)	(0, 7)	(0, 7)	(0, 7)	(0, 3)	(0, 3)	(0, 0)	(0, 1)
2^3	(0, 12)	(0, **17**)	(0, 13)	(0, 13)	(0, 11)	(0, 8)	(0, 4)	(0, 2)	(0, 1)
2^4	(0, 19)	(0, 23)	(0, **26**)	(0, 23)	(0, 22)	(0, 19)	(0, 14)	(0, 8)	(0, 1)
2^5	(0, 29)	(0, 42)	(0, **44**)	(1, **44**)	(0, **44**)	(0, 40)	(0, 32)	(0, 16)	(0, 5)
2^6	(1, 46)	(1, 58)	(0, **60**)	(1, **60**)	(0, **60**)	(0, 59)	(0, 55)	(0, 44)	(0, 22)
2^7	(1, **60**)	(2, **60**)	(1, **60**)	(1, **60**)	(0, **60**)	(0, **60**)	(1, **60**)	(0, **60**)	(0, 52)

4.2. Comparison with l_1-PSVM

We compare l_p-PSVM with algorithm l_1-PSVM in this part, and if $p = 1$ the problem (8) turns to be a convex problem

$$min_{z \in R^{n+1}} \|Az - e\|_2^2 + \lambda \|z\|_1. \quad (18)$$

For every dataset, we use 5-fold cross-validation error to choose the appropriate parameters $(\bar{\lambda}, \bar{p})$ for l_p-PSVM and $\tilde{\lambda}$ for algorhtm l_1-PSVM, the following table gives out the numerical results, where $\bar{\sharp}$ mean the number of zero variables in $\bar{w}$ of algorithm l_p-PSVM determined by bound (13), $\tilde{\sharp}$ means the number of zero variables in $\tilde{w}$ of algorithm l_1-PSVM.

Table 5: Numerical results

dataset \ algorithm	l_p-PSVM	l_1-PSVM
heart	$79.63\%(\bar{\lambda} = 2^6, \bar{p} = 0.3, \bar{\sharp} = 8)$	$79.63\%(\tilde{\lambda} = 1, \tilde{\sharp} = 3)$
Australian	$85.8\%(\bar{\lambda} = 2^5, \bar{p} = 0.3, \bar{\sharp} = 6)$	$85.94\%(\tilde{\lambda} = 128, \tilde{\sharp} = 2)$
Sonar	$77.51\%(\bar{\lambda} = 2^6, \bar{p} = 0.2, \bar{\sharp} = 58)$	$75.62\%(\tilde{\lambda} = 16, \tilde{\sharp} = 36)$
German	$75.7\%(\bar{\lambda} = 2^7, \bar{p} = 0.3, \bar{\sharp} = 13)$	$76.1\%(\tilde{\lambda} = 4, \tilde{\sharp} = 13)$

From Table 5 we find that l_p-PSVM successes in finding more sparse solution with higher accuracy than or almost the same with l_1-PSVM.

4.3. Credit Card Dataset

Now we test the performance of l_p-PSVM on credit card dataset. The 6000 credit card records used in this paper were selected from 25,000 real-life credit card records of a major US bank. Each record has 113 columns or variables to describe the cardholders behaviors, including balance, purchases, payment cash advance and so on. With the accumulated experience functions, we eventually get 65 variables from the original 113 variables to describe the cardholders' behaviors[15, 17, 19].

In this paper we chose the holdout method on credit card dataset to separate data into training set and testing set: first, the bankruptcy dataset (960 records) is divided into 10 intervals (each interval has approximately 100 records). Within each interval, 25 records are randomly selected. Thus the total of 250 bankruptcy records is obtained after repeating 10 times. Then, as the same way, we get 250 current records from the current dataset. Finally, the total of 250 bankruptcy records and 250 current records are combined to form a single training dataset, with the remaining 710 lost records and 4790 current records merge into a testing dataset. This process is performed for five times, and for each time we apply l_p-PSVM to training and test, and in each training, we apply 5-fold cross-validation to choose appropriate parameters in l_p-PSVM for testing. Here, we apply three scores to evaluate two algorithms: sensitivity (Sn), specificity(Sp) and G-Mean(g)on

$$Sn = \frac{TP}{TP + FN}, \tag{19}$$

$$Sp = \frac{TN}{TN + FP}, \tag{20}$$

$$g = \sqrt{Sn \times Sp}, \tag{21}$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative. At last we recorded the corresponding average scores for each time in Table 6, where p^*, λ^* are the optimal parameters corresponding to higher average g of 5-fold cross-validation; $\bar{\sharp}$ means the number of zero variables in $\bar{w}$ of algorithm l_p-PSVM determined by bound (13). We can see that l_p-PSVM actually finds sparse features and gets high performance scores.

In this experiment, we also prove that: for a given p, with the increase of λ, solution turns to be more sparse; for a given λ, more sparse solution appears in the interval $p \in [0.2, 0.6]$, which give us practical guide in choosing appropriate parameters.

5. Conclusions

We proposed an efficient model which combines proximal support vector machine (PSVM) and feature selection strategy by introducing the l_p-norm $(0 < p < 1)$ in its primal problem. Based on the theoretical results of [10], two

Table 6: Experiments on Credit Card Dataset

	Training Set						Testing Set		
	Sp	Sn	g	p^*	λ^*	$\sharp$	Sn	Sp	g
DS 1	74.26%	83.57%	78.22%	0.3	2^6	56	74.15%	78.98%	76.53%
DS 2	67.85%	83.16%	74.74%	0.4	2^6	55	68.97%	85.9%	76.97%
DS 3	71.88%	80.73%	75.69%	0.2	2^6	56	73.15%	83.07%	77.95%
DS 4	71.07%	82.33%	76.33%	0.3	2^6	58	73.19%	80.68%	76.84%
DS 5	70.66%	80.32%	74.85%	0.1	2^7	56	73.92%	80.39%	77.09%

lower bounds for the absolute value of nonzero entries in every local optimal solution of l_p-PSVM is also developed. After smoothing the problem in l_p-norm PSVM and solving it by smoothing conjugate gradient method (SCG) method, preliminary numerical experiments show the effectiveness of algorithm l_p-PSVM. Further development of SVMs with l_p norm may be in two ways: Introducing kernel functions to solve nonlinear classification problem by l_p-PSVM; Develop other lower bounds for the absolute value of nonzero entries in every local optimal solution of l_p- norm standard SVMs, such as C-SVM, ν-SVM and etc.

Acknowledgments

This work has been partially supported by grants from National Natural Science Foundation of China(NO.10601064, NO.70621001).

References

[1] B. Boser, I. Guyon and V. Vapnik, A training algorithm for optimal margin classifiers, Computational Learing Theory, 144-152, 1992.

[2] V. Vapnik, Statistical Learning Theory. New York, NY:Wiley, 1998.

[3] J. Friedman and T. Hastie(eds.), Discussion of "Consistency in boosting" by W. Jiang, G. Lugosi, N. Vayatis and T. Zhang, Ann. Statist. 32,102-107, 2004.

[4] J. Zhu, S. Rosset, T. Hastie and R. Tibshirani , 1-norm Advances in Neural Information Processing Systems 16, 2004.

[5] I. Guyon , J. Weston , S. Barnhill and V. Vapnik, Gene selection for cancer classification using support vector machines, Machine Learning 46, 389-422, 2002.

[6] J. Weston, S. Mukherjee and V. Vapnik(eds.), Feature selection for svms. Advances in Neural Information Processing Systems 13, 2001.

[7] P. Bradley and O. Mangasarian, Feature selection via concave minimization and support vector machines, International Conference on Machine Learning, Morgan Kaufmann, 1998.

[8] M. Song, C. Breneman, J. Bi , N. Sukumar, K. Bennett, S. Cramer and Tugcu N(2002). Prediction of protein retention times in anion-exchange chromatography systems using support vector regression. J. Chemical Information and Computer Sciences.

[9] H. Zou and M. Yuan, The f_∞ norm support vector machine. Statistica Sinica, 18, 379-398, 2008.

[10] Chen X, Xu F and Ye Y. Lower Bound Theory of Nonzero Entries in Solutions of l_2-l_p Minimization. Technical report, Department of Applied Mathematics, The Hong Kong Polytechnic University.

[11] X. Chen and W. Zhou, Smoothing nonlinear conjugate gradient method for image restoration using nonsmooth nonconvex minimization, Preprint, Department of Ap- plied Mathematics, The Hong Kong Polytechnic University, 2008.

[12] G. Fung and O. Mangasarian, Proximal support vector machine classifiers, Proceedings of International Conference of Knowledge Discovery and Data Mining, 77-86, 2001.

[13] Y. Chen and C. Lin, Combining SVMs with various feature selection strategies. Feature Extraction, Foundations and Applications. New York, Springer, 319-328, 2005.

[14] V. Vapnik , O. Chapelle, Bounds on error expectation for SVM. In Advances in Large-Margin Classifiers (Neural Information Processing), chapter 14, MIT press, 261-280, 2000.

[15] Y. Shi, Y. Peng, W. X. Xu and X. W. Tang, Data mining via multiple criteria linear programming: applications in credit card portfolio management. International Journal of Information Technology and Decision Making, 131-151, 2002

[16] L.G. Zhou, K.K.Lai and J. Yen, credit scoring models with AUC maxinization based on weighted SVM, International Journal of Information Technology and Decision Making, 677-696, 2009.

[17] J. He, X.T. Liu and Y. Shi(eds.), Classifications of cr edit cardholder behavior by using fuzzy linear programming, International Journal of Information Technology and Decision Making, 633-650, 2004.

[18] L. Yu, S. Y. Wang and J. Cao, A modified least squares support vector machine classifier with application, International Journal of Information Technology and Decision Making, 697-710, 2009.

[19] Y. Shi, Current research trend: information technology and decision making in 2008, International Journal of Information Technology and Decision Making, 1-5, 2009.